Presentación Equipo 2

De La Cruz Chavarria Christian Israel

Ramírez Vázquez Ezequiel Alberto

Mariscal Viorato Maria Fernanda

Oscar Beltran Villegas

Fase 1

- El viernes 21 de mayo quedaron asignados nuestros lugares en el equipo en esta primera fase, quedando como Christian con el código, Alberto con la documentación, Oscar con el código en Python y a mi con la administración. El mismo viernes se comenzó a realizar la primera fase, creando los primeros cuatros módulos comenzando con un orden de izquierda a derecha del single datapath que son PC, Add, la memoria de instrucciones y un multiplexor de dos a uno. Lo primero en realizarse fue el PC, en seguida el Ader después la memoria de instrucciones, la unidad de control el multiplexor y el banco de registros, así sucesivamente. En ese primer día se tiene un registro de 5 errores aproximadamente y no se termino de completar la ALU y la ALU Control. Este primer día se lograron completar alrededor de 7 módulos. Posteriormente a este día se estuvieron realizando algunos cambios al código por los errores antes mencionados. Se registraron errores en el PC ya que no se tenia inicializada la entrada la cual era la que iba a permitir que diera el primer Cero para que se comenzara a leer la memoria de instrucciones . Los Multiplexores en si no tuvieron mucha complicidad ya que los volvió una ternaria para que en el primer caso le entregara el valor necesario cambiando los ceros y los unos por las entradas a las que estaba conectado el multiplexor.
- El sábado 22 de mayo se realizo una reunión en Discord, para solucionar una duda que se tenia con el código para la alu, en la multiplicación y la división. Uno de los errores que se presentaron en este día y que se solucionario entre este día y el domingo fue que se recibía únicamente x en los lugares en donde se supone que debía haber una respuesta (esto se generaba en el banco de registro). Otro error generado fue que en la posición No. 2 se generaba un numero negativo; Esto además que se escribía únicamente una vez, se generaba en una posición errónea. Todos estos errores se manejaban de manera secuencial , por lo que se tuvieron que haces cambios de cada modulo para que cada modulo sea por blocking.

- Domingo 23 de mayo: surgieron dudas con la realización de Single Datapath, sobre si crear un modulo para la AND sin embargo se pudo solucionar haciéndolo directamente en Single Datapath. Otro error que genero problemas este día, fue la condición IF del banco de registros la cual permite escribir (esta condición te dice si puedes o no hacerlo dependiendo de donde este conectado). Sin embargo al momento dela compilación el TopLevel se si clava y su solución fue meter la dirección a esta misma conexión del IF esto con la finalidad de volver a la dirección un valor valido. En este caso un valor valido seria 1 sin embargo, esto seria necesario cambiarlo, por que nosotros no estamos poniendo un valor en esto, así que este procedimiento tiene un poco de lógica.

- **Nota:** El código fue terminado supuestamente la madrugada del domingo, concluyendo con un Tb y se realizaron todas las instancias correctamente y estos fueron subidos a la plataforma de Github el día Lunes 24 por la tarde noche. Para poder lograr el desarrollo de este código se utilizaron varias herramientas y fuentes de información, la principal, el libro de la clase y el de MIPS. Sin embargo esto no fue así, ya que en el trascurso de los días, el código volvió a probarse y se observo un error que antes no había sido percatado por ninguno de los integrantes del equipo, quedando concluido correctamente el día 26 de mayo.

- Lunes 24 de mayo: Fue nuestra reunión de equipo en Discord para aclarar algunas dudas, y explicación del código, nos reunimos con la finalidad de que nos mostrara paso por paso lo que hizo como lo hizo y el porque lo hizo para poder así nosotros desarrollar las demás actividades que teníamos pendientes y poder poner en práctica lo que mi compañero Christian había elaborado para esta elaboración. Además el mismo lunes se comenzó con la elaboración del código en Python, en este caso mi compañero Oscar decidió elaborarlo en un lenguaje en C.

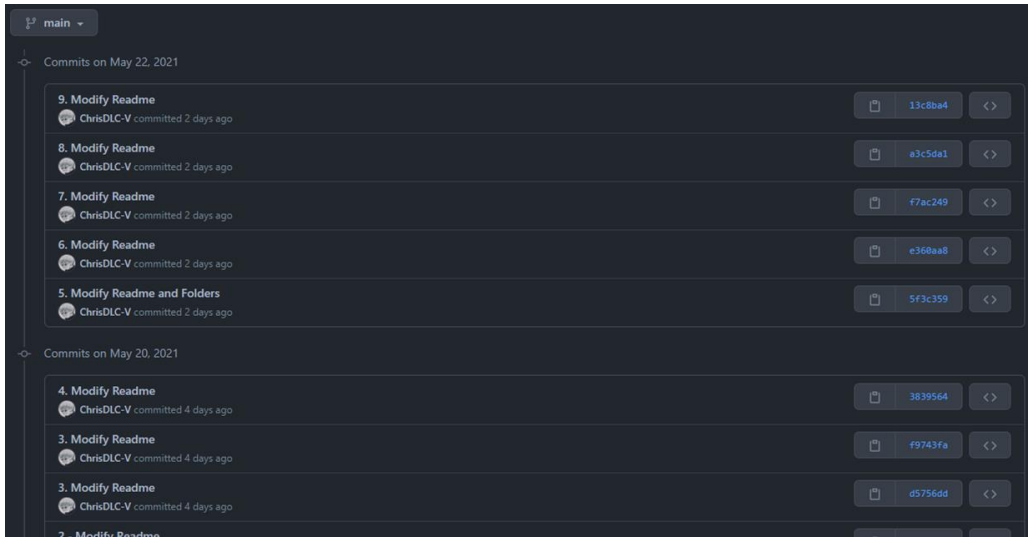
- El decodificador de lenguaje ensamblador a código máquina está desarrollado en lenguaje C de computadora, compuesto principalmente por oraciones condicionales y convertidor de números decimales a binarios. Primero, se crea un menú y las opciones para agregar, eliminar e imprimir la última opción también finalizan el procedimiento. Lo primero en el programa es imprimir los comandos que se han ingresado juntos en el menú, pero si no hay comandos y aún no aparecen, solo aparece el menú, luego en el código seguirá pidiendo comandos y configurando condiciones para la operación, y luego configure los datos rt, rt y rd Simplemente guarde, y luego, envíe los datos rt, rt y rd a una función que los convierta en números binarios, y el contenido devuelto los convertirá en formato de caracteres para poder agregar "0" a la izquierda, porque estos no están reservados en números enteros.

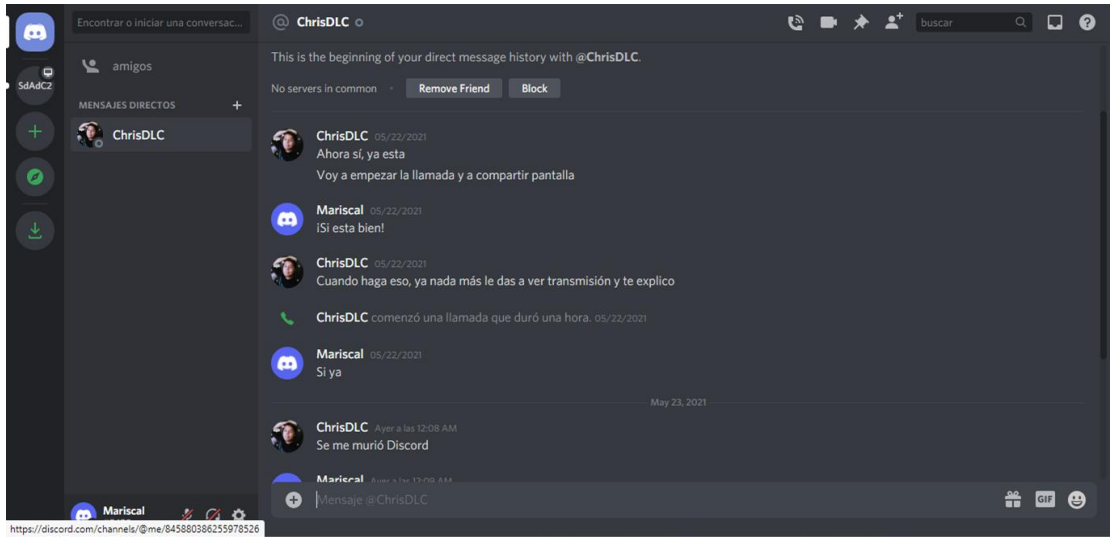
- Se utilizaron las condiciones de la operación solicitada para obtener una función binaria. Una vw < op con "000000" y shamt se ha inicializado con "00000", para que podamos combinar todas las instrucciones juntas. Nada más concatena todo con el tipo de variable char con 32 espacios en su orden respectivo. Y todas las operaciones se pueden realizar hasta 32 veces, porque el tamaño de esta configuración es lo que el usuario desea, pero siempre que se cambie el valor de la constante, todo se puede expandir. Para ello, se crea un bucle do while, en el que se encuentra toda la estructura del programa, y finaliza cuando la opción de salida es preciosa. El trabajo que se realiza antes de salir es hacer que el programa obtenga todas las instrucciones del binario y crea el texto. El principal problema que hubo fue el uso de cuerdas. La primera es comparar una cadena de if, por ejemplo, poner if (varChar == "add") in, no da ningún resultado, pero el problema que veo es que no puede ser tan simple como eso y después de un período de tiempo En el proceso de investigación comparativa, encontré una función llamada strcmp de la biblioteca string.h, que se usa para comparar la cadena requerida con otra cadena, y luego conservar la parte. ya teniendo el numero en binario en un numero entero falta agregarle los "0" faltantes a la izquierda y para eso también fue un problema y también tarde mucho en encontrar como hacerle, pero me encontré una función que me ayudo a matar dos pájaros de un tiro ya que lo que yo buscaba primero era convertir el numero entero a una cadena de caracteres y después de eso agregarles los "0" a la izquierda, y esta función que encontré hacia las dos cosas porque esta lo que hacía era imprimir tal cual como le quisieras mandar el dato como un printf pero este se llama fprintf es casi lo mismo pero este tiene un tercer apartado, que puedes donde imprimir lo que tu deseas, ejemplo fprintf(varOut, "hola mundo %d", intBinario); y con eso se guardó correctamente el dato en cadena de caracteres. Esto ayudo mucho a conocer el fprintf ya que se utilizo para imprimir las instrucciones en el archivo de texto ya que se daba la elección de elegir que se imprimieras ahí.

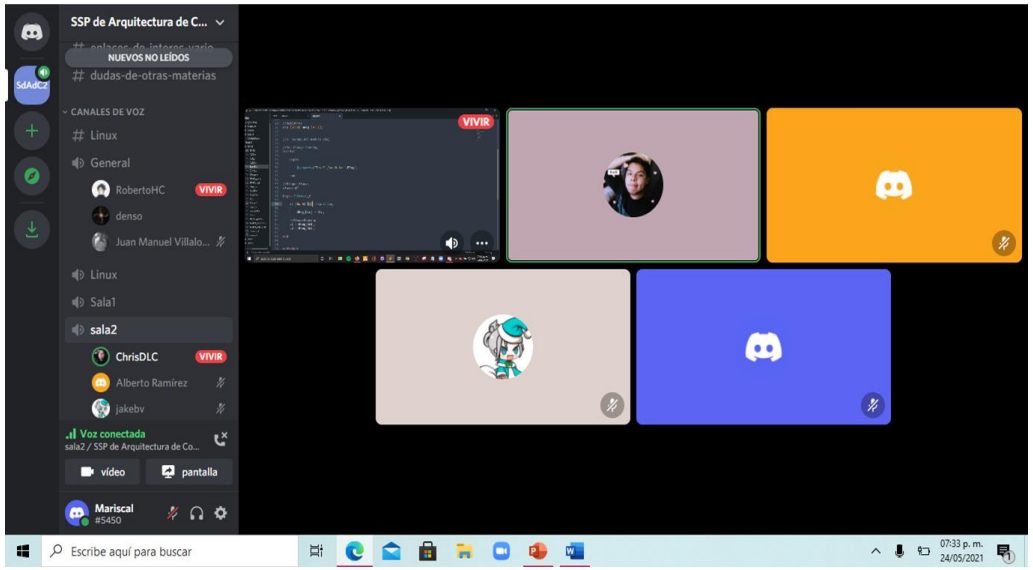
- Miercoles 26 de Mayo: Se realizo una tercera reunión en disrcod ya que surgió un imprevisto con el código, y aun faltaban muchas cosas por ordenar en las demás áreas. El equipo se conecto desde las 7:00 pm hasta las 10:50 pm para poder resolver un error a la hora de compilar, ya que si bien algunos resultados estaban correctos, hacían falta dos nueros de la tabla que no aparecían en esta compilación, así que nos reunimos y entre todos comenzamos a revisar el código, creándole nuevos módulos, quitándole y poniéndole “begin”, “case” entre otras cosas. Se decidió revisar el código modulo por modulo, y uno de los compañeros se dio cuenta que un comando estaba mal escrito poniendo una “w” en lugar de una “B”, se modifico únicamente eso, y el código compilo a la perfección. Ese mismo día se surgieron dudas también con la documentación, dudas que por la hora ya no podían aclararse.

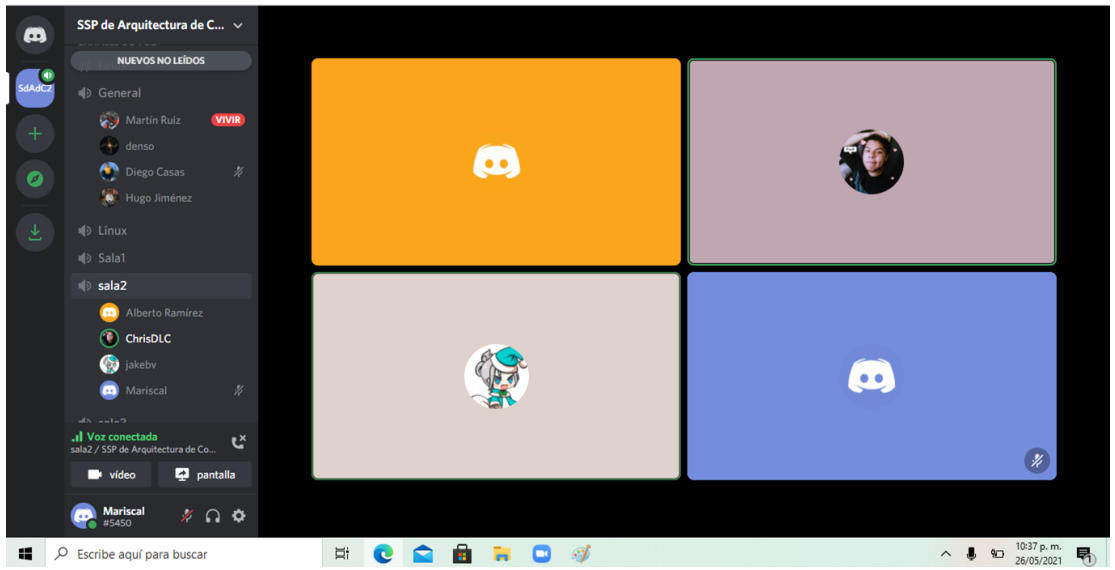
- Jueves 27 de mayo: Se aclaro la duda en la documentación de la cual se trataba sobre uno de los puntos de desarrollo en uno de los temas, se también este día se termino de pasar toda la información requerida tanto para la documentación como el reporte.

Evidencias


main
Commits on May 22, 2021
9. Modify Readme
ChrisDLC-V committed 2 days ago
13c8ba4
8. Modify Readme
ChrisDLC-V committed 2 days ago
a3c5da1
7. Modify Readme
ChrisDLC-V committed 2 days ago
f7ac249
6. Modify Readme
ChrisDLC-V committed 2 days ago
e360aa8
5. Modify Readme and Folders
ChrisDLC-V committed 2 days ago
5f3c359
Commits on May 20, 2021
4. Modify Readme
ChrisDLC-V committed 4 days ago
3839564
3. Modify Readme
ChrisDLC-V committed 4 days ago
f9743fa
3. Modify Readme
ChrisDLC-V committed 4 days ago
d5756dd


Encontrar o iniciar una conversac...
ChrisDLC
buscar
amigos
MENSAJES DIRECTOS
ChrisDLC
This is the beginning of your direct message history with @ChrisDLC.
No servers in common
Remove Friend
Block
ChrisDLC 05/22/2021
Ahora sí, ya esta
Voy a empezar la llamada y a compartir pantalla
Mariscal 05/22/2021
¡Si esta bien!
ChrisDLC 05/22/2021
Cuando haga eso, ya nada más le das a ver transmisión y te explico
ChrisDLC comenzó una llamada que duró una hora. 05/22/2021
Mariscal 05/22/2021
Si ya
May 23, 2021
ChrisDLC Ayer a las 12:08 AM
Se me murió Discord
Mariscal
Mensaje @ChrisDLC
https://discord.com/channels/@me/845880386255978526


SSP de Arquitectura de C...
NUEVOS NO LEÍDOS
dudas-de-otras-materias
CANALES DE VOZ
Linux
General
RobertoHC
VIVIR
denso
Juan Manuel Villalo...
Linux
Sala1
sala2
ChrisDLC
VIVIR
Alberto Ramírez
jakebv
Voz conectada
sala2 / SSP de Arquitectura de Co...
vídeo
pantalla
Mariscal
#5450
Escribe aquí para buscar
07:33 p. m.
24/05/2021


SSP de Arquitectura de C...
NUEVOS NO LEÍDOS
General
Martín Ruiz
VIVIR
denso
Diego Casas
Hugo Jiménez
Linux
Sala1
sala2
Alberto Ramírez
ChrisDLC
jakebv
Mariscal
Voz conectada
sala2 / SSP de Arquitectura de Co...
vídeo
pantalla
Mariscal
#5450
Escribe aquí para buscar
10:37 p. m.
26/05/2021